Equipo 2

1850231 Cid Sanabria Dulce Ximena

1851895 Ortiz Cruz Jaclyn Lizeth

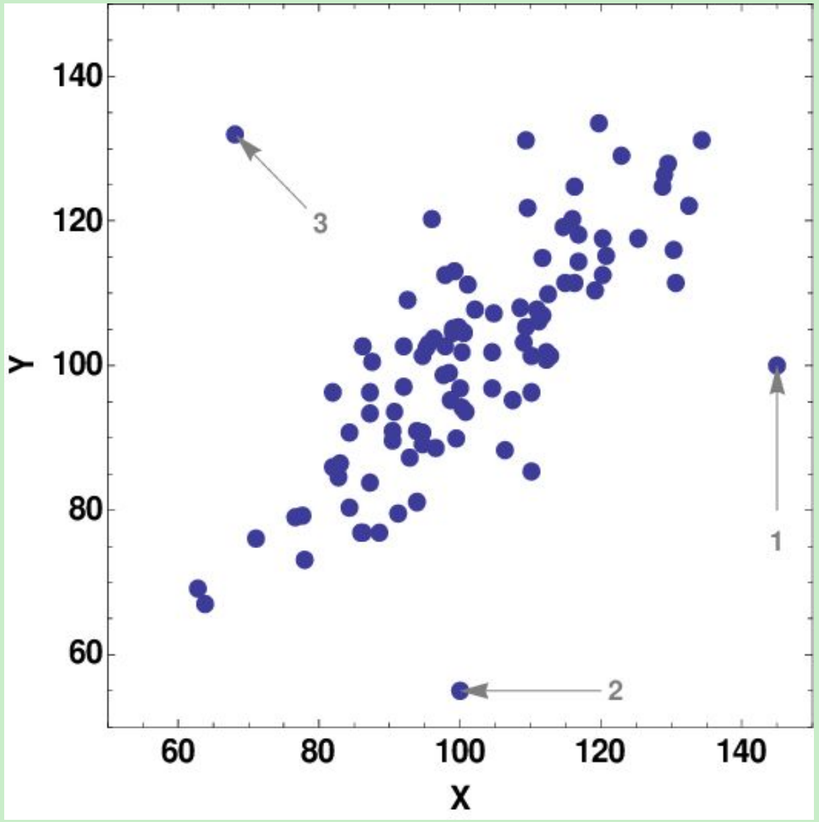
1849687 Rincón Pacheco Diego Alejandro

1854410 Serrano Caballero Lizeth

Detección de Bordes

Outliers

¿Qué son los outliers?



Es una observación en una muestra estadística o serie temporal que parece estar fuera del lugar, el cual afecta potencialmente a la estimación de los parámetros del mismo.

Deben ser contemplados en el contexto del análisis y debe evaluarse el tipo de información que pueden proporcionar.

01

« Un outlier es una observación que se desvía tanto de otras observaciones que despierta la sospecha de haber sido generado por un mecanismo diferente »

- Hawkings, 1980

02

« Una observación (o subconjunto de observaciones) que parecen ser inconsistentes respecto del resto de ese conjunto de datos »

- Barnet y otros, 1994

¿Cómo son causados?

1. Errores de procedimiento
2. Errores en la construcción de la base de datos
3. Acontecimientos extraordinarios
4. Valores extremos
5. Causas no conocidas

Tipos de outliers

Dependiendo en la distribución subyacente de los datos, un outlier puede ser una de las siguientes tipos:

01 Un valor extremo o relativamente extremo.

02 Un contaminante, valor de otra distribución, posiblemente desconocida.

03 Un valor legítimo pero inesperado.

04 Un valor que ha sido medido o grabado mal.

Cómo detectarlos

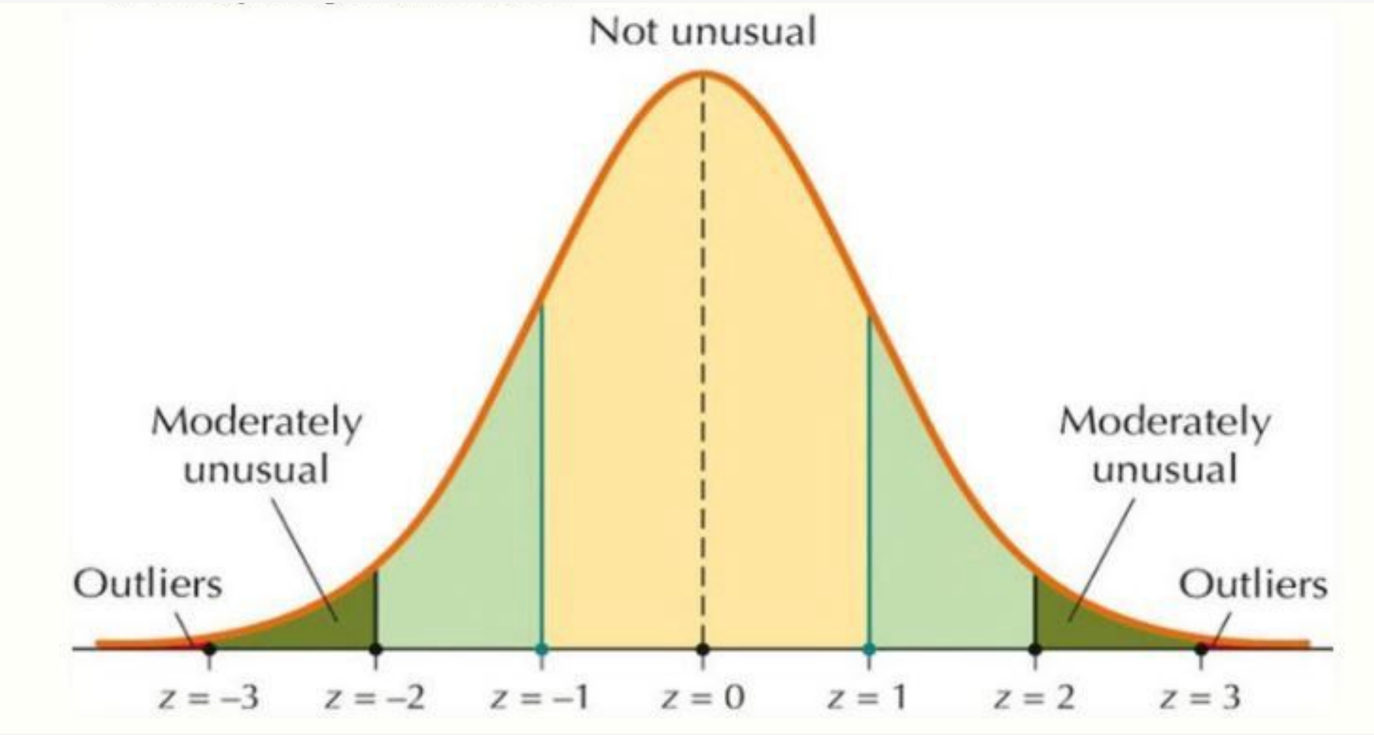
Cuando el conjunto de datos es pequeño, podemos detectar el valor atípico con solo mirar los datos. ¿Qué pasa si tenemos una gran base de datos, cómo identificamos los valores atípicos?

1 **Z-score**

2 **Boxplot**

3 **IQR**

Z-score



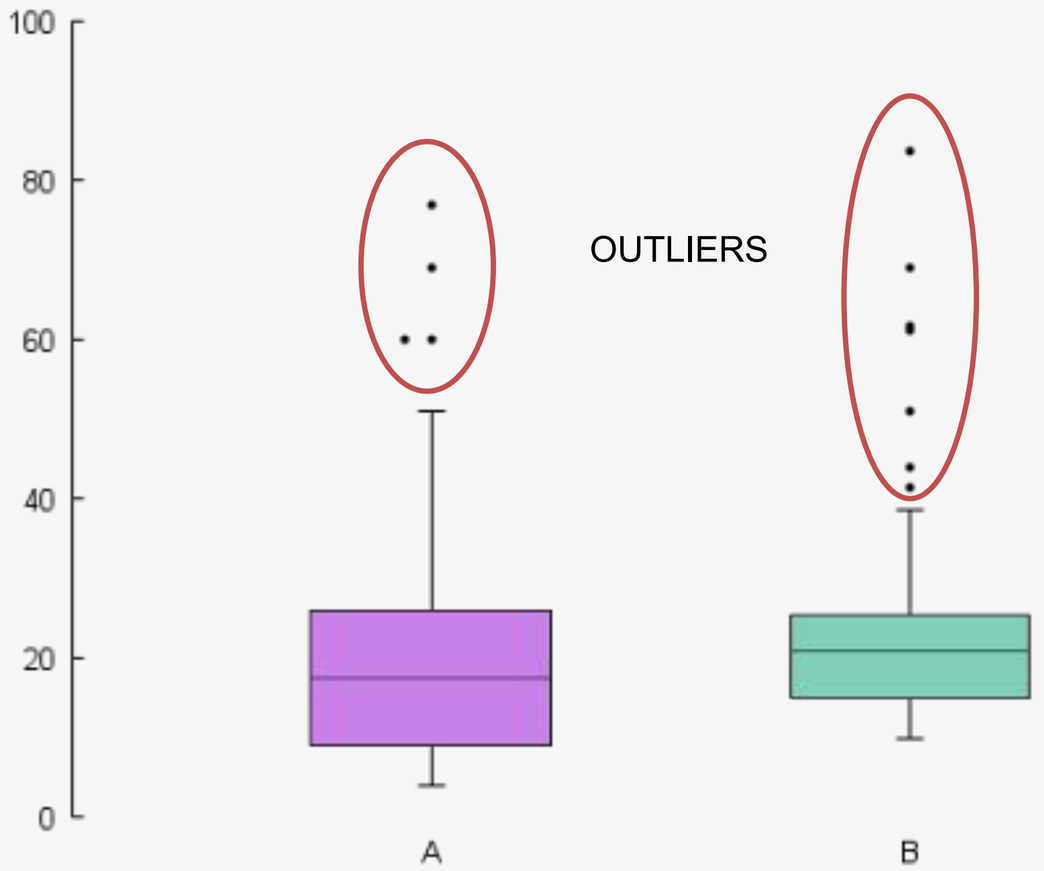
Número de desviaciones estándar por las cuales el valor de una observación está por encima del valor medio de lo que se está observando o midiendo.

Cualquier punto que sea más de 3 veces la desviación estándar, es muy probable que sea atípico.

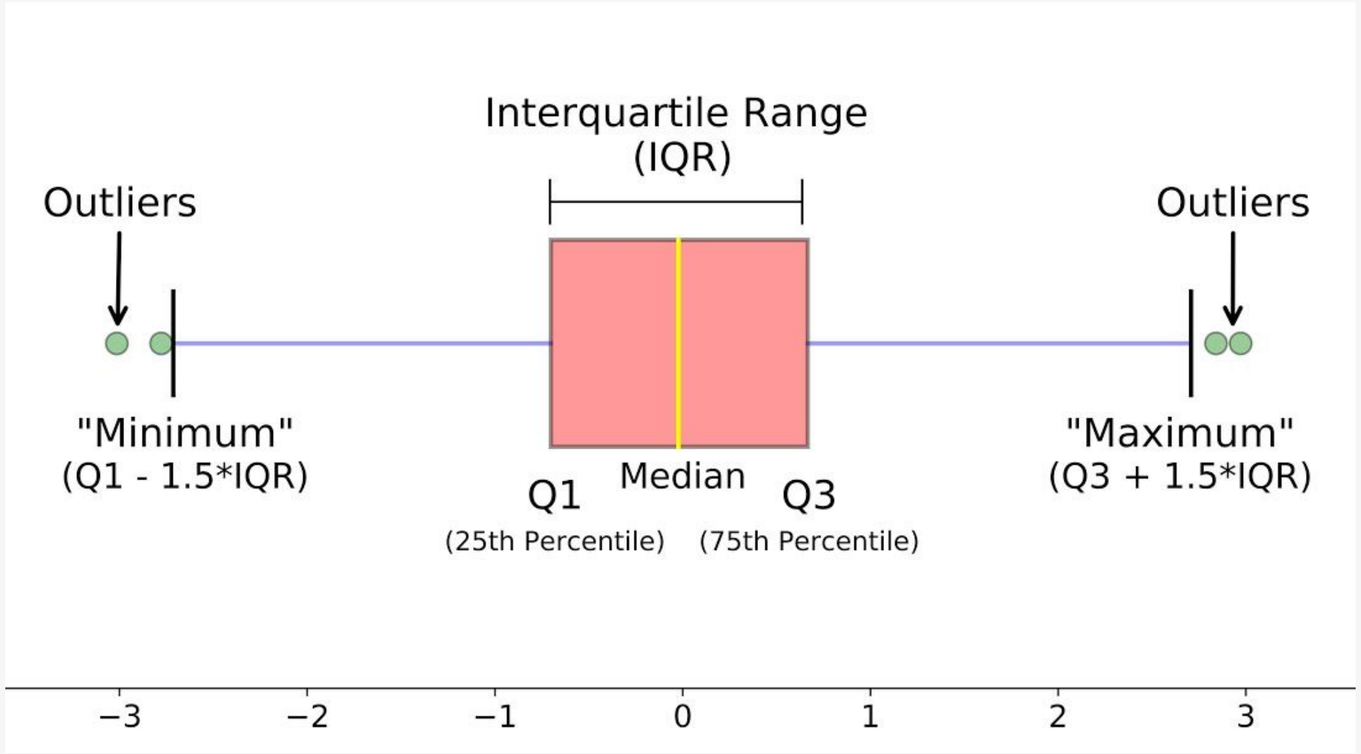
Boxplot

Representación gráfica de datos numéricos a través de sus cuantiles.

Cualquier punto de datos que se muestre por encima o por debajo de los límites puede considerarse como outlier.



IQR



Se utiliza para definir los valores atípicos

Es la diferencia entre el tercer cuartil y el primer cuartil. Los valores atípicos son aquellas observaciones que se encuentran debajo de (Q1 - 1.5*IQR) o arriba de (Q3 + 1.5*IQR).

Cómo Manejarlos

Una vez que hemos detectado los outliers, nuestra siguiente pregunta debe ser que vamos a hacer con ellos. Para esto tenemos las siguientes opciones:

01 Mantener los outliers

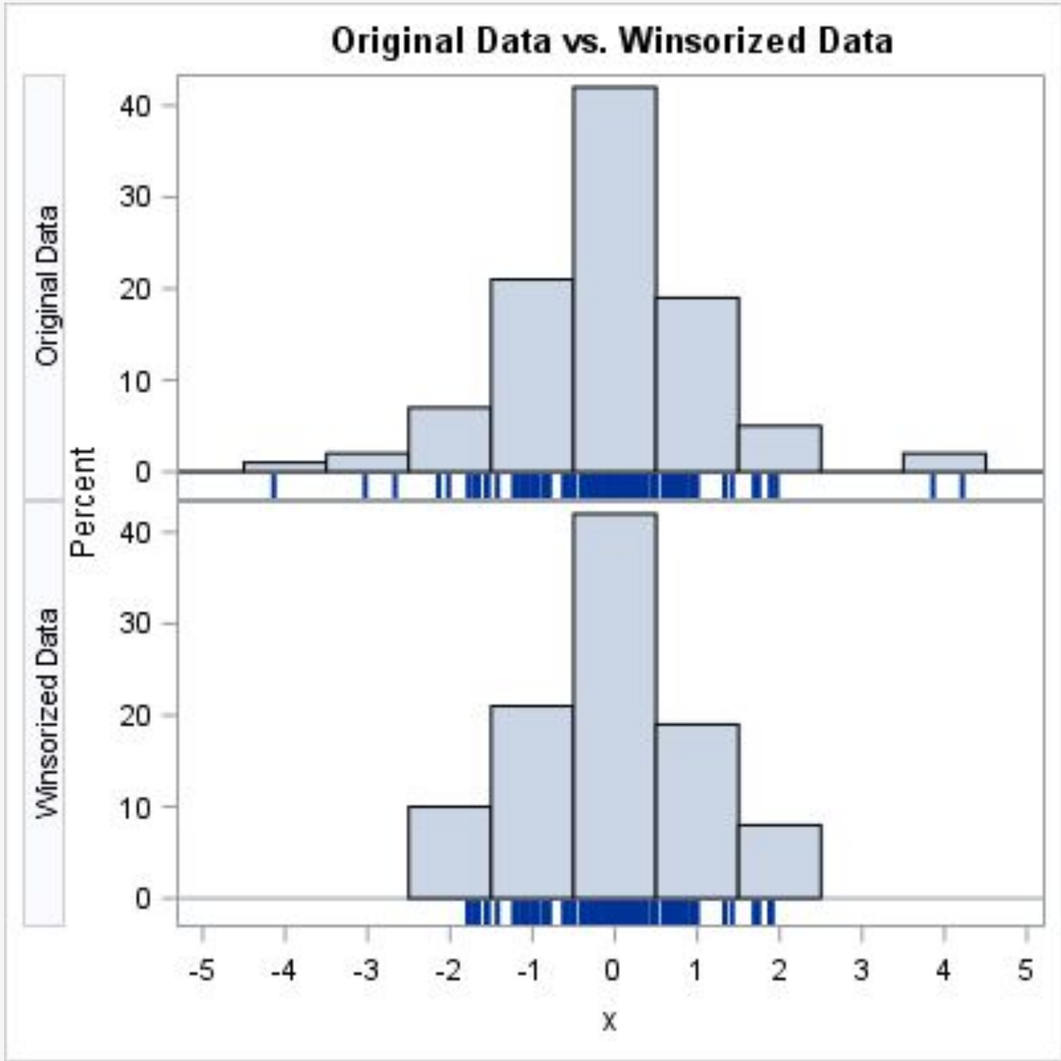
02 Remover los outliers

03 Winsorizing

04 Imputación en base a media/mediana

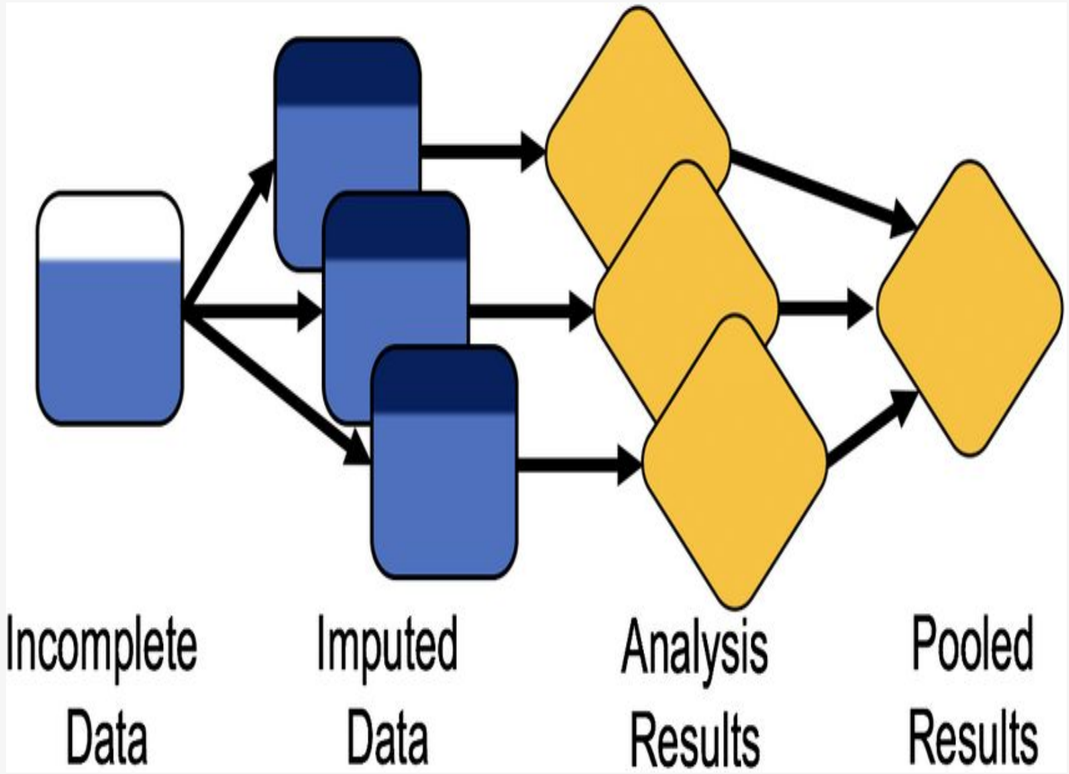
Winsorizing

Consiste en que cada valor por encima del límite percentil superior o debajo del límite percentil inferior serán cambiados por el valor que se encuentra en su correspondiente límite percentil.



Imputación

En este proceso, los outliers son reemplazados por los valores de la mediana de los datos, ya que los outliers pueden tener mucha influencia en la media



Ejercicio Iris

El primer paso es importar las librerías que se usarán en el proceso. *Seaborn* es una librería que genera fácilmente diferentes gráficos.

```
[1]  import seaborn as sns
     import pandas as pd
     import numpy as np
```

Se abre la base de datos iris y se conocen los datos que contiene.

```
[2]  iris=sns.load_dataset("iris")
```

```
[3]  iris.shape

     (150, 5)
```

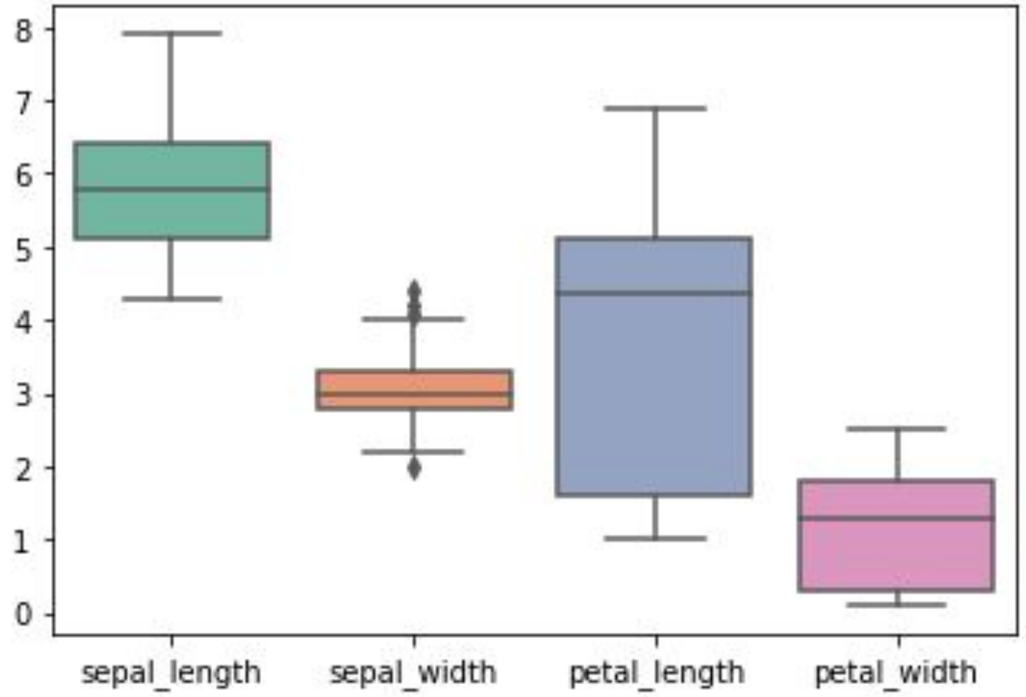
```
[4]  iris.head()
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Ejercicio Iris

A continuación se muestra el diagrama de caja con la ayuda de la función **boxplot**.

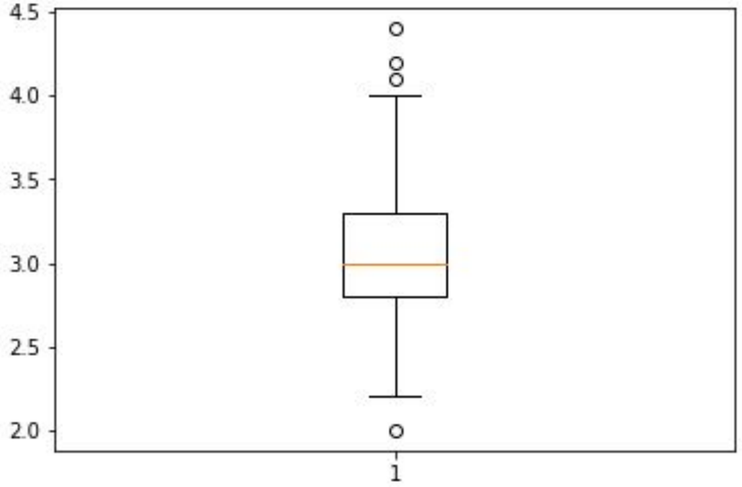
```
[5]  bp = sns.boxplot(data=iris, orient="v", palette="Set2")
```




```
[7]  import matplotlib.pyplot as plt
```

```
plt.boxplot(iris["sepal_width"])
```

```
{'boxes': [<matplotlib.lines.Line2D at 0x7fde75d85210>],
 'caps': [<matplotlib.lines.Line2D at 0x7fde75d8c250>,
  <matplotlib.lines.Line2D at 0x7fde75d8c790>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fde75d97290>],
 'means': [],
 'medians': [<matplotlib.lines.Line2D at 0x7fde75d8cd10>],
 'whiskers': [<matplotlib.lines.Line2D at 0x7fde76ad6250>,
  <matplotlib.lines.Line2D at 0x7fde75d85cd0>]}
```



Ejercicio Iris

Después de observar que hay presencia de datos atípicos, es hora de saber sus valores.

```
[6]  outliers = []
     def outliers_iqr(data):
         data = sorted(data)
         q1 = np.percentile(data, 25)
         q3 = np.percentile(data, 75)
         IQR = q3-q1 #calcula el iqr
         lim_inf = q1-(1.5*IQR) #limite inf
         lim_sup = q3+(1.5*IQR) #limite sup
         # Ciclo para saber cuales son atípicos
         for i in data:
             if (i<lim_inf or i>lim_sup):
                 outliers.append(i)
         return outliers
     out = outliers_iqr(iris["sepal_width"])
     print("Los outliers por el método IQR son: ", out)
```

```
Los outliers por el método IQR son:  [2.0, 4.1, 4.2, 4.4]
```

Ejercicio Iris

Winsorizing

Antes de trabajar con el proceso Winsorize, es recomendable realizar una copia de la base de datos para que sea más ordenado.

```
[74] bd_win = bd.copy(deep=True) #crea una copia de la base de datos para trabajar en ella
```

Posteriormente se importa winsorize desde la librería Scipy.

```
[75] from scipy.stats.mstats import winsorize
```

Con la librería cargada, se procede a usar la función winsorize para poder transformar los outliers como se desee.

```
[76] bd_win['SepalW_w_5%'] = winsorize(bd['sepal_width'], limits=(0.05, 0.05))#winsorizing con k=5 (5% y 95%)
```

Ejercicio Iris

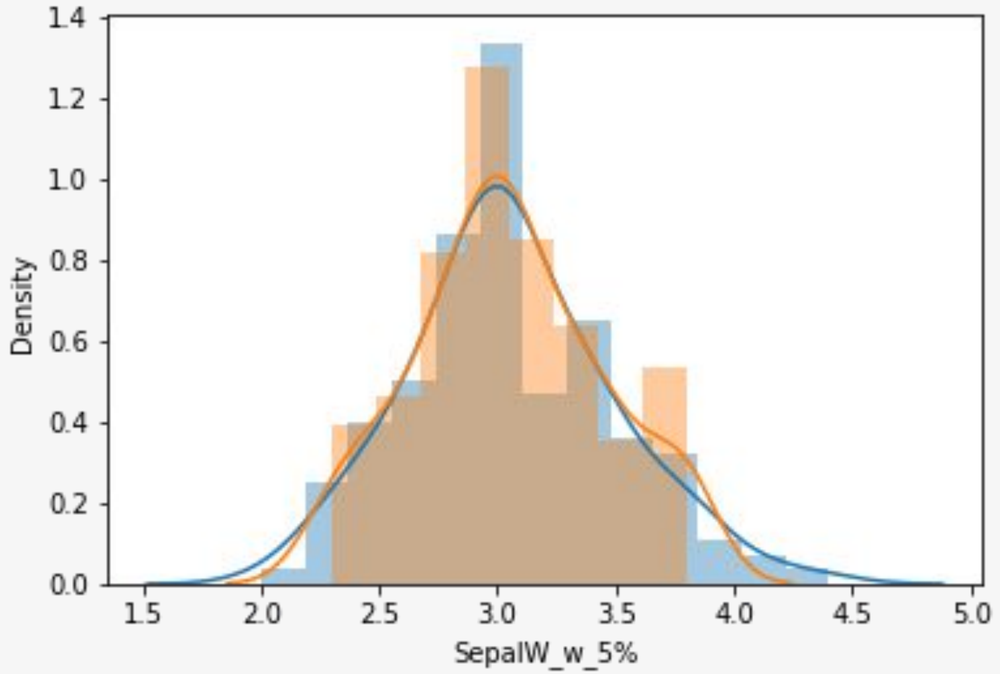
Si se quieren saber los valores con los que estará trabajando el proceso, se puede realizar lo siguiente con ayuda de la librería numpy.

```
[153] noventaycinco_pct = np.percentile(bd.sepal_width, 95)
      cinco_pct = np.percentile(bd.sepal_width, 5)
      print("valor del 95%:",noventaycinco_pct)
      print("valor del 5%:",cinco_pct)
```

```
valor del 95%: 3.8
valor del 5%: 2.3449999999999998
```

```
#Distribución sin winsorize
sns.distplot(bd['sepal_width']) #plot en color azul

#Nueva Distribución con Winsorize
sns.distplot(bd_win['SepalW_w_5%']) #plot en color naranja
```



Ejercicio Iris

Imputación media/mediana

Lo primero que se debe realizar es obtener la mediana de los datos.

```
median=np.median(bd.sepal_width)
median
```

```
3.0
```

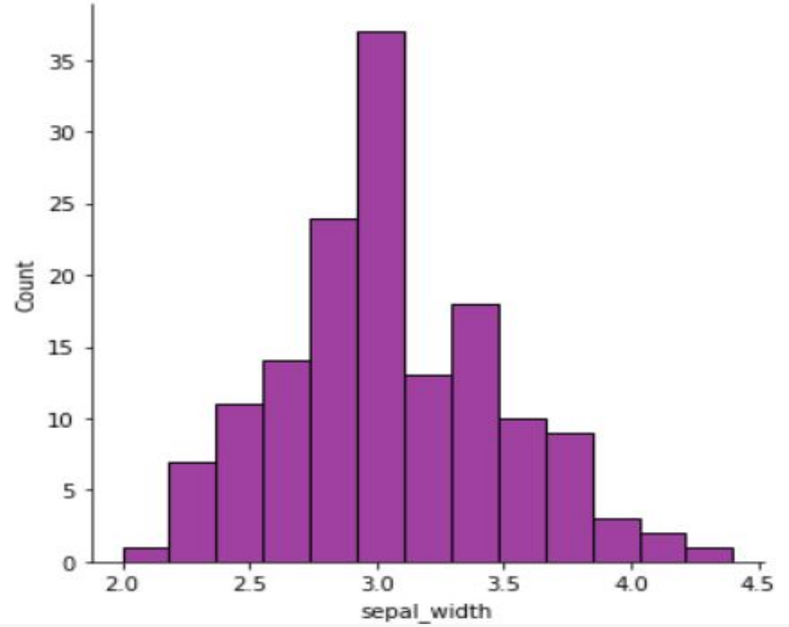
Ya con el valor de la mediana, se utiliza la función where de Numpy para poder cambiar los valores outliers por el valor de la mediana.

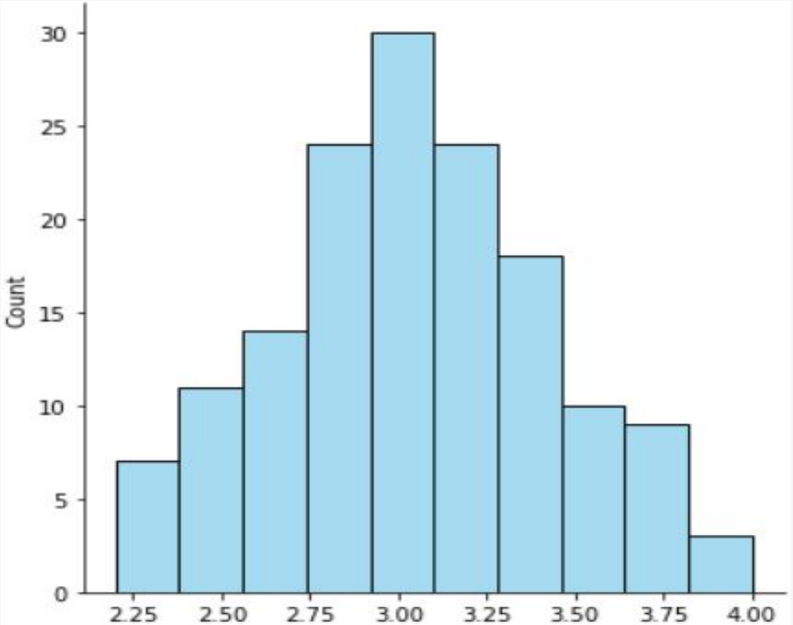
```
q1 = np.percentile(bd.sepal_width, 25)
q3 = np.percentile(bd.sepal_width, 75)
IQR = q3-q1 #calcula el iqr
lim_inf = q1-(1.5*IQR) #limite inf
lim_sup = q3+(1.5*IQR) #limite sup
print("limites inferior y superior:",lim_inf,lim_sup)

a=np.where((bd.sepal_width > lim_sup) | (bd.sepal_width < lim_inf),median,bd.sepal_width)
# where funciona como un if else; el símbolo | es como un or.
print(a)
```

Ejercicio Iris

```
limites inferior y superior: 2.05 4.05
[3.5 3.  3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 3.7 3.4 3.  3.  4.  3.  3.9 3.5
 3.8 3.8 3.4 3.7 3.6 3.3 3.4 3.  3.4 3.5 3.4 3.2 3.1 3.4 3.  3.  3.1 3.2
 3.5 3.6 3.  3.4 3.5 2.3 3.2 3.5 3.8 3.  3.8 3.2 3.7 3.3 3.2 3.2 3.1 2.3
 2.8 2.8 3.3 2.4 2.9 2.7 3.  3.  2.2 2.9 2.9 3.1 3.  2.7 2.2 2.5 3.2 2.8
 2.5 2.8 2.9 3.  2.8 3.  2.9 2.6 2.4 2.4 2.7 2.7 3.  3.4 3.1 2.3 3.  2.5
 2.6 3.  2.6 2.3 2.7 3.  2.9 2.9 2.5 2.8 3.3 2.7 3.  2.9 3.  3.  2.5 2.9
 2.5 3.6 3.2 2.7 3.  2.5 2.8 3.2 3.  3.8 2.6 2.2 3.2 2.8 2.8 2.7 3.3 3.2
 2.8 3.  2.8 3.  2.8 3.8 2.8 2.8 2.6 3.  3.4 3.1 3.  3.1 3.1 3.1 2.7 3.2
 3.3 3.  2.5 3.  3.4 3. ]
```





Preguntas

01 ¿Qué se debe hacer en caso de observar un dato outliers?

02 ¿Cómo identificamos los valores atípicos?

03 ¿Cómo se obtienen los valores atípicos mediante IQR?

04 ¿En qué consiste la transformación de las estadísticas “Winsorizing?

05 Menciona 3 aplicaciones de la detección de outliers

Conclusión

Encontrar outliers es una tarea importante para muchas aplicaciones ya que, si son tomados en cuenta, pueden alterar de forma significativa nuestros resultados y crear interpretaciones engañosas de los mismos. Por esto, es importante aprender a identificarlos y sobre todo aprender a trabajar con ellos.

Bibliografía

Sharma, N. (2021, August 3). Ways to detect and remove the outliers. Medium.
https://towardsdatascience.com/ways-to-detect-and-remove-the-outliers-404d16608dba

Analytics Vidhya. (2021, May 21). Detecting and treating outliers: How to handle outliers. Analytics Vidhya.
https://www.analyticsvidhya.com/blog/2021/05/detecting-and-treating-outliers-treating-the-odd-one-out/

Ichi Pro (2020, December 20). 5 formas de detectar valores atípicos que todo científico de datos debería conocer (Código python). ICHI.PRO.
https://ichi.pro/es/5-formas-de-detectar-valores-atipicos-que-todo-cientifico-de-datos-deberia-conocer-codigo-python-123855099504163

Adrián Alfredo de Armas (2015, febrero 23). Detección de outliers en grandes bases de datos.UADE
https://repositorio.uade.edu.ar/xmlui/bitstream/handle/123456789/2520/De%20Armas.pdf?sequence=1&isAllowed=y

Francisco M. Ocaña Peinado.Tratamiento de outliers y missing.
https://www.ugr.es/~fmocan/MATERIALES%20DOCTORADO/Tratamiento%20de%20outliers%20y%20missing.pdf

Alicia Horsch (2014, Feb) Detecting and Treating Outliers in Python-Part 3
https://towardsdatascience.com/detecting-and-treating-outliers-in-python-part-3-dcb54abaf7b0